

모두의 핏 앱 API 명세서

1. 문서 개요

1.1 프로젝트 정보

- **프로젝트명:** 모두의 핏 (All4Fit) 앱
- **API 버전:** v1
- **Base URL:** <https://api.all4fit.co.kr/v1>
- **인증 방식:** JWT Bearer Token
- **응답 형식:** JSON
- **작성일:** 2025년 10월

1.2 공통 응답 형식

성공 응답

```
{  
    "success": true,  
    "message": "요청이 성공적으로 처리되었습니다.",  
    "data": {  
        // 응답 데이터  
    }  
}
```

에러 응답

```
{  
    "success": false,  
    "message": "에러 메시지",  
    "error": {  
        "code": "ERROR_CODE",  
        "details": "상세 에러 정보"  
    }  
}
```

1.3 HTTP 상태 코드

- **200:** 성공
- **201:** 생성 성공
- **400:** 잘못된 요청
- **401:** 인증 실패
- **403:** 권한 없음
- **404:** 리소스 없음

- 409: 충돌 (중복 등)
- 500: 서버 오류

2. 인증 API

2.1 통합 로그인 (LOGIN_UNIFIED)

POST /auth/login

이메일/전화번호와 비밀번호로 로그인합니다.

요청 헤더:

```
Content-Type: application/json
```

요청 본문:

```
{
  "loginId": "user@example.com",
  "password": "password123!"
}
```

요청 매개변수:

필드명	타입	필수	설명
loginId	string	O	이메일 또는 전화번호
password	string	O	비밀번호

응답 본문:

```
{
  "success": true,
  "message": "로그인이 성공했습니다.",
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "expiresIn": 3600,
    "user": {
      "userId": 123,
      "email": "user@example.com",
      "nickname": "길동이",
      "userType": "USER",
      "profileImage": "https://example.com/profile.jpg"
    }
  }
}
```

2.2 카카오 로그인 (LOGIN_KAKAO)

POST /auth/kakao

카카오 계정으로 로그인합니다.

요청 본문:

```
{  
    "accessToken": "kakao_access_token"  
}
```

요청 매개변수:

필드명	타입	필수	설명
accessToken	string	O	카카오 액세스 토큰

응답 본문:

```
{  
    "success": true,  
    "message": "카카오 로그인이 성공했습니다.",  
    "data": {  
        "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
        "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",  
        "expiresIn": 3600,  
        "user": {  
            "userId": 123,  
            "email": "user@example.com",  
            "nickname": "길동이",  
            "userType": "USER",  
            "profileImage": "https://example.com/profile.jpg",  
            "isNewUser": false  
        }  
    }  
}
```

2.3 비밀번호 찾기 (PW_FIND)

POST /auth/forgot-password

이름과 전화번호로 비밀번호 재설정을 요청합니다.

요청 본문:

```
{  
  "name": "홍길동",  
  "phone": "010-1234-5678"  
}
```

요청 매개변수:

필드명	타입	필수	설명
name	string	O	가입한 이름
phone	string	O	가입한 전화번호

응답 본문:

```
{  
  "success": true,  
  "message": "SMS 인증번호가 발송되었습니다.",  
  "data": {  
    "verificationId": "verification_123",  
    "expiresIn": 300  
  }  
}
```

POST /auth/verify-phone

SMS 인증번호를 확인합니다.

요청 본문:

```
{  
  "verificationId": "verification_123",  
  "code": "123456"  
}
```

요청 매개변수:

필드명	타입	필수	설명
verificationId	string	O	인증 요청 ID
code	string	O	SMS 인증번호 (6자리)

응답 본문:

```
{  
  "success": true,  
  "message": "인증이 완료되었습니다.",  
  "data": {  
    "verified": true,  
    "purposeId": "purpose_123"  
  }  
}
```

POST /auth/reset-password

새 비밀번호를 설정합니다.

요청 본문:

```
{  
  "purposeId": "purpose_123",  
  "newPassword": "newpassword123!"  
}
```

요청 매개변수:

필드명	타입	필수	설명
purposeId	string	O	인증 완료 후 받은 목적 ID
newPassword	string	O	새 비밀번호 (8자 이상, 영문+숫자+특수문자)

응답 본문:

```
{  
  "success": true,  
  "message": "비밀번호가 성공적으로 변경되었습니다."  
}
```

2.4 일반 사용자 회원가입 (SIGNUP)

POST /auth/register

일반 사용자 회원가입을 처리합니다.

요청 본문:

```
{  
  "email": "user@example.com",  
  "phone": "010-1234-5678",  
  "password": "SecurePass123!"  
}
```

```

"password": "password123!",
"name": "홍길동",
"nickname": "길동이",
"birthDate": "1990-01-01",
"gender": "M",
"address": "서울시 강남구 테헤란로 123",
"userType": "USER",
"termsAgreed": {
  "service": true,
  "privacy": true,
  "marketing": false
}
}

```

요청 매개변수:

필드명	타입	필수	설명
email	string	O	이메일 주소
phone	string	O	전화번호 (010-xxxx-xxxx 형식)
password	string	O	비밀번호 (8자 이상, 영문+숫자+특수문자)
name	string	O	실명
nickname	string	O	닉네임 (2-20자)
birthDate	string	O	생년월일 (YYYY-MM-DD)
gender	enum	O	성별 (M, F, OTHER)
address	string	X	주소
userType	enum	O	사용자 유형 (USER, INSTRUCTOR, OPERATOR)
termsAgreed	object	O	약관 동의 정보
termsAgreed.service	boolean	O	서비스 이용약관 동의
termsAgreed.privacy	boolean	O	개인정보 처리방침 동의
termsAgreed.marketing	boolean	X	마케팅 정보 수신 동의

응답 본문:

```
{
  "success": true,
  "message": "회원가입이 완료되었습니다.",
  "data": {
    "userId": 123,
    "email": "user@example.com",
    "nickname": "길동이",
    "userType": "USER",
    "status": "ACTIVE",
  }
}
```

```
        "createdAt": "2024-10-01T00:00:00Z"
    }
}
```

2.5 지도자 회원가입 (INSTRUCTOR_SIGNUP)

POST /auth/instructor/register

지도자 회원가입을 처리합니다.

요청 본문:

```
{
  "basicInfo": {
    "email": "instructor@example.com",
    "phone": "010-1234-5678",
    "password": "password123!",
    "name": "김지도자",
    "nickname": "김지도자",
    "birthDate": "1985-01-01",
    "gender": "M"
  },
  "instructorInfo": {
    "businessNumber": "123-45-67890",
    "certification": "생활체육지도사 2급",
    "experience": "10년간 헬스장에서 근무",
    "specializations": ["헬스", "필라테스"],
    "hourlyRate": 50000
  },
  "facilityInfo": {
    "facilityName": "강남 헬스장",
    "facilityAddress": "서울시 강남구 테헤란로 123",
    "postalCode": "06292",
    "latitude": 37.5665,
    "longitude": 126.9780
  },
  "termsAgreed": {
    "service": true,
    "privacy": true,
    "marketing": false
  }
}
```

요청 매개변수:

필드명	타입	필수	설명
basicInfo	object	O	기본 회원 정보
basicInfo.email	string	O	이메일 주소

필드명	타입	필수	설명
basicInfo.phone	string	O	전화번호
basicInfo.password	string	O	비밀번호
basicInfo.name	string	O	실명
basicInfo.nickname	string	O	닉네임
basicInfo.birthDate	string	O	생년월일
basicInfo.gender	enum	O	성별 (M, F, OTHER)
instructorInfo	object	O	지도자 정보
instructorInfo.businessNumber	string	O	사업자 번호
instructorInfo.certification	string	O	자격증
instructorInfo.experience	string	O	경력
instructorInfo.specializations	array	O	전문 분야
instructorInfo.hourlyRate	number	O	시간당 요금
facilityInfo	object	O	소속 시설 정보
facilityInfo.facilityName	string	O	시설명
facilityInfo.facilityAddress	string	O	시설 주소
facilityInfo.postalCode	string	X	우편번호
facilityInfo.latitude	number	O	위도
facilityInfo.longitude	number	O	경도
termsAgreed	object	O	약관 동의 정보

참고: 지도자가 회원가입할 때 시설 정보를 등록하면, 해당 시설의 지도자 탭에서 이름과 한줄소개만 표시됩니다. 지도자 상세 페이지는 없습니다.

2.6 시설 운영자 회원가입 (FACILITY_OPERATOR_SIGNUP)

POST /auth/operator/register

시설 운영자 회원가입을 처리합니다.

요청 본문:

```
{
  "basicInfo": {
    "email": "operator@example.com",
    "phone": "010-1234-5678",
    "password": "password123!",
    "name": "김운영자",
  }
}
```

```

"nickname": "김운영자",
"birthDate": "1980-01-01",
"gender": "M"
},
"businessInfo": {
  "representativeName": "김운영자",
  "businessNumber": "123-45-67890"
},
"operatingInfo": {
  "operatingHours": {
    "monday": "06:00-24:00",
    "tuesday": "06:00-24:00",
    "wednesday": "06:00-24:00",
    "thursday": "06:00-24:00",
    "friday": "06:00-24:00",
    "saturday": "08:00-22:00",
    "sunday": "08:00-22:00"
  },
  "holidays": "매월 둘째, 넷째 일요일"
},
"documents": {
  "businessLicenseUrl": "https://example.com/business_license.jpg"
},
"termsAgreed": {
  "service": true,
  "privacy": true,
  "marketing": false
}
}
}

```

요청 매개변수:

필드명	타입	필수	설명
basicInfo	object	O	기본 회원 정보
basicInfo.email	string	O	이메일 주소
basicInfo.phone	string	O	전화번호
basicInfo.password	string	O	비밀번호
basicInfo.name	string	O	실명
basicInfo.nickname	string	O	닉네임
basicInfo.birthDate	string	O	생년월일
basicInfo.gender	enum	O	성별 (M, F, OTHER)
businessInfo	object	O	사업자 정보
businessInfo.representativeName	string	O	대표자명
businessInfo.businessNumber	string	O	사업자 번호

필드명	타입	필수	설명
operatingInfo	object	O	운영 정보
operatingInfo.operatingHours	object	O	요일별 운영시간
operatingInfo.holidays	string	X	휴관일
documents	object	O	서류
documents.businessLicenseUrl	string	O	사업자 등록증 URL
termsAgreed	object	O	약관 동의 정보

2.7 토큰 갱신

POST /auth/refresh

액세스 토큰을 갱신합니다.

요청 헤더:

```
Authorization: Bearer refresh_token
```

응답 본문:

```
{
  "success": true,
  "message": "토큰이 갱신되었습니다.",
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "expiresIn": 3600
  }
}
```

2.8 로그아웃

POST /auth/logout

사용자를 로그아웃 처리합니다.

요청 헤더:

```
Authorization: Bearer access_token
```

응답 본문:

```
{  
  "success": true,  
  "message": "로그아웃이 완료되었습니다."  
}
```

2.9 중복 확인 API

POST /auth/check-email

이메일 중복 확인을 합니다.

요청 본문:

```
{  
  "email": "user@example.com"  
}
```

응답 본문:

```
{  
  "success": true,  
  "data": {  
    "available": true,  
    "message": "사용 가능한 이메일입니다."  
  }  
}
```

POST /auth/check-phone

전화번호 중복 확인을 합니다.

요청 본문:

```
{  
  "phone": "010-1234-5678"  
}
```

POST /auth/check-nickname

닉네임 중복 확인을 합니다.

요청 본문:

```
{  
  "nickname": "길동이"  
}
```

2.10 전화번호 인증 API

POST /auth/send-verification

SMS 인증번호를 발송합니다.

요청 본문:

```
{  
  "phone": "010-1234-5678",  
  "type": "SIGNUP"  
}
```

요청 매개변수:

필드명	타입	필수	설명
phone	string	O	전화번호
type	enum	O	인증 유형 (SIGNUP, PASSWORD_RESET, PHONE_CHANGE)

응답 본문:

```
{  
  "success": true,  
  "message": "인증번호가 발송되었습니다.",  
  "data": {  
    "verificationId": "verification_123",  
    "expiresIn": 300  
  }  
}
```

3. 사용자 관리 API

3.1 프로필 조회 (MYPAGE_MAIN)

GET /users/profile

현재 사용자의 프로필 정보를 조회합니다.

요청 헤더:

```
Authorization: Bearer access_token
```

응답 본문:

```
{
  "success": true,
  "data": {
    "userId": 123,
    "email": "user@example.com",
    "phone": "010-1234-5678",
    "name": "홍길동",
    "nickname": "길동이",
    "birthDate": "1990-01-01",
    "gender": "M",
    "address": "서울시 강남구 테헤란로 123",
    "userType": "USER",
    "status": "ACTIVE",
    "profileImage": "https://example.com/profile.jpg",
    "bio": "안녕하세요! 운동을 좋아합니다.",
    "createdAt": "2024-10-01T00:00:00Z",
    "lastLoginAt": "2024-10-01T12:00:00Z"
  }
}
```

3.2 프로필 수정 (MYPAGE_PROFILE_EDIT)

PUT /users/profile

사용자 프로필 정보를 수정합니다.

요청 헤더:

```
Authorization: Bearer access_token
Content-Type: application/json
```

요청 본문:

```
{
  "nickname": "새로운닉네임",
  "bio": "수정된 자기소개",
  "address": "서울시 서초구 서초대로 456"
}
```

요청 매개변수:

필드명	타입	필수	설명
nickname	string	X	닉네임
bio	string	X	자기소개
address	string	X	주소

3.3 프로필 이미지 업로드

POST /users/profile/image

프로필 이미지를 업로드합니다.

요청 헤더:

```
Authorization: Bearer access_token
Content-Type: multipart/form-data
```

요청 본문:

```
image: [이미지 파일]
```

응답 본문:

```
{
  "success": true,
  "message": "프로필 이미지가 업로드되었습니다.",
  "data": {
    "profileImageUrl": "https://example.com/new-profile.jpg"
  }
}
```

3.4 계정 정보 변경 (MYPAGE_ACCOUNT_CHANGE)

PUT /users/account/phone (MYPAGE_PHONE_CHANGE)

전화번호를 변경합니다.

요청 본문:

```
{
  "currentPassword": "current_password",
  "newPhone": "010-9876-5432",
  "verificationCode": "123456"
}
```

요청 매개변수:

필드명	타입	필수	설명
currentPassword	string	O	현재 비밀번호
newPhone	string	O	새 전화번호
verificationCode	string	O	SMS 인증번호

PUT /users/account/email (MYPAGE_EMAIL_CHANGE)

이메일을 변경합니다.

요청 본문:

```
{  
  "currentPassword": "current_password",  
  "newEmail": "newemail@example.com",  
  "verificationCode": "123456"  
}
```

PUT /users/account/password (MYPAGE_PASSWORD_CHANGE)

비밀번호를 변경합니다.

요청 본문:

```
{  
  "currentPassword": "current_password",  
  "newPassword": "newpassword123!"  
}
```

3.5 회원 탈퇴 (MYPAGE_WITHDRAW_CONFIRM)**DELETE /users/account**

회원 탈퇴를 처리합니다.

요청 본문:

```
{  
  "password": "current_password",  
  "reason": "서비스 불만족"  
}
```

요청 매개변수:

필드명	타입	필수	설명
password	string	O	현재 비밀번호
reason	string	X	탈퇴 사유

4. 시설 관련 API

4.1 시설 목록 조회 (FACILITY_LIST)

GET /facilities

시설 목록을 조회합니다.

쿼리 매개변수:

매개변수	타입	필수	기본값	설명
page	integer	X	1	페이지 번호
limit	integer	X	20	페이지당 항목 수
keyword	string	X		검색 키워드
facilityType	string	X		시설 유형 (GYM, POOL, TENNIS, etc.)
sportId	integer	X		운동 종목 ID
latitude	float	X		사용자 위도
longitude	float	X		사용자 경도
radius	integer	X	5000	검색 반경 (미터)
sortBy	string	X	distance	정렬 기준 (distance, rating, price)
sortOrder	string	X	asc	정렬 순서 (asc, desc)

응답 본문:

```
{
  "success": true,
  "data": [
    {
      "facilityId": 1,
      "name": "강남 헬스장",
      "description": "최신 시설을 갖춘 헬스장입니다.",
      "address": "서울시 강남구 테헤란로 123",
      "latitude": 37.5665,
      "longitude": 126.9780,
      "phone": "02-1234-5678",
      "facilityType": "GYM",
      "averageRating": 4.5,
      "lastUpdate": "2023-10-01T10:00:00Z"
    }
  ]
}
```

```

    "reviewCount": 128,
    "distance": 500,
    "mainImage": "https://example.com/facility1.jpg",
    "sports": [
        {
            "sportId": 1,
            "name": "헬스",
            "pricePerHour": 15000
        }
    ],
    "amenities": ["주차장", "샤워실", "락커룸"]
},
],
"pagination": {
    "page": 1,
    "limit": 20,
    "total": 100,
    "totalPages": 5
}
}
}

```

4.2 시설 상세 조회 (FACILITY_DETAIL)

GET /facilities/{facilityId}

특정 시설의 상세 정보를 조회합니다.

경로 매개변수:

매개변수	타입	설명
facilityId	integer	시설 ID

응답 본문:

```
{
    "success": true,
    "data": {
        "facilityId": 1,
        "name": "강남 헬스장",
        "description": "최신 시설을 갖춘 헬스장입니다.",
        "address": "서울시 강남구 테헤란로 123",
        "latitude": 37.5665,
        "longitude": 126.9780,
        "phone": "02-1234-5678",
        "email": "contact@gym.com",
        "website": "https://gym.com",
        "facilityType": "GYM",
        "status": "ACTIVE",
        "averageRating": 4.5,
        "reviewCount": 128,
    }
}
```

```

"images": [
  {
    "imageId": 1,
    "imageUrl": "https://example.com/facility1-1.jpg",
    "altText": "헬스장 내부",
    "sortOrder": 1
  }
],
"sports": [
  {
    "sportId": 1,
    "name": "헬스",
    "description": "다양한 운동기구를 이용한 전신 운동",
    "pricePerHour": 15000,
    "isAvailable": true
  }
],
"operatingHours": {
  "monday": "06:00-24:00",
  "tuesday": "06:00-24:00",
  "wednesday": "06:00-24:00",
  "thursday": "06:00-24:00",
  "friday": "06:00-24:00",
  "saturday": "08:00-22:00",
  "sunday": "08:00-22:00"
},
"amenities": ["주차장", "샤워실", "락커룸", "사우나"],
"createdAt": "2024-01-01T00:00:00Z"
}
}

```

5. 운동 종목 API

5.1 운동 종목 목록 조회

GET /sports

운동 종목 목록을 조회합니다.

쿼리 매개변수:

매개변수	타입	필수	기본값	설명
category	string	X		카테고리 필터
isActive	boolean	X	true	활성 상태 필터

응답 본문:

```
{
  "success": true,
```

```

"data": [
  {
    "sportId": 1,
    "name": "헬스",
    "description": "다양한 운동기구를 이용한 전신 운동",
    "iconUrl": "https://example.com/icons/gym.png",
    "category": "FITNESS",
    "isActive": true
  },
  {
    "sportId": 2,
    "name": "수영",
    "description": "전신 근력과 심폐 기능 향상",
    "iconUrl": "https://example.com/icons/swimming.png",
    "category": "AQUATIC",
    "isActive": true
  }
]
}

```

6. 검색 API

6.1 시설 검색 (HOME_SEARCH)

GET /search/facilities

시설을 검색합니다.

쿼리 매개변수:

매개변수	타입	필수	기본값	설명
keyword	string	O		검색 키워드
latitude	float	X		사용자 위도
longitude	float	X		사용자 경도
radius	integer	X	5000	검색 반경 (미터)
page	integer	X	1	페이지 번호
limit	integer	X	20	페이지당 항목 수

응답 본문:

```
{
  "success": true,
  "data": [
    {
      "facilityId": 1,
      "name": "강남 헬스장",
      "address": "123 강남대로, 강남구, 서울",
      "lat": 37.5502,
      "lon": 127.0437,
      "category": "GYM"
    }
  ]
}
```

```
        "address": "서울시 강남구 테헤란로 123",
        "latitude": 37.5665,
        "longitude": 126.9780,
        "distance": 500,
        "mainImage": "https://example.com/facility1.jpg",
        "averageRating": 4.5,
        "facilityType": "GYM"
    }
],
"pagination": {
    "page": 1,
    "limit": 20,
    "total": 50,
    "totalPages": 3
}
}
```

6.2 검색 기록 조회

GET /search/history

사용자의 검색 기록을 조회합니다.

요청 헤더:

```
Authorization: Bearer access_token
```

쿼리 매개변수:

매개변수	타입	필수	기본값	설명
page	integer	X	1	페이지 번호
limit	integer	X	20	페이지당 항목 수

응답 본문:

```
{
    "success": true,
    "data": [
        {
            "searchId": 1,
            "keyword": "헬스",
            "searchType": "FACILITY",
            "createdAt": "2024-10-01T10:00:00Z"
        }
    ],
    "pagination": {
        "page": 1,
        "limit": 20,
        "total": 50,
        "totalPages": 3
    }
}
```

```
        "limit": 20,  
        "total": 25,  
        "totalPages": 2  
    }  
}
```

7. 위치 관련 API

7.1 사용자 위치 설정 (HOME_LOCATION_CHANGE)

POST /users/location

사용자의 위치를 설정합니다.

요청 헤더:

```
Authorization: Bearer access_token  
Content-Type: application/json
```

요청 본문:

```
{  
    "address": "서울시 강남구 테헤란로 123",  
    "latitude": 37.5665,  
    "longitude": 126.9780,  
    "isDefault": true  
}
```

요청 매개변수:

필드명	타입	필수	설명
address	string	O	주소
latitude	number	O	위도
longitude	number	O	경도
isDefault	boolean	X	기본 위치 여부

7.2 사용자 위치 목록 조회

GET /users/locations

사용자의 저장된 위치 목록을 조회합니다.

요청 헤더:

```
Authorization: Bearer access_token
```

응답 본문:

```
{
  "success": true,
  "data": [
    {
      "locationId": 1,
      "address": "서울시 강남구 테헤란로 123",
      "latitude": 37.5665,
      "longitude": 126.9780,
      "isDefault": true,
      "createdAt": "2024-10-01T00:00:00Z"
    }
  ]
}
```

8. 지도 관련 API

8.1 지도 시설 조회 (MAP_TAB)

GET /map/facilities

지도 영역 내의 시설을 조회합니다.

쿼리 매개변수:

매개변수	타입	필수	설명
northEastLat	float	O	북동쪽 위도
northEastLng	float	O	북동쪽 경도
southWestLat	float	O	남서쪽 위도
southWestLng	float	O	남서쪽 경도
facilityType	string	X	시설 유형 필터
zoomLevel	integer	X	줌 레벨

응답 본문:

```
{
  "success": true,
  "data": [
    {
      "facilityId": 1,
      "name": "设施名称"
    }
  ]
}
```

```
        "name": "강남 헬스장",
        "address": "서울시 강남구 테헤란로 123",
        "latitude": 37.5665,
        "longitude": 126.9780,
        "facilityType": "GYM",
        "averageRating": 4.5,
        "mainImage": "https://example.com/facility1.jpg"
    }
]
}
```

9. 에러 코드

9.1 인증 관련 에러

코드	HTTP 상태	설명
AUTH_001	401	인증 토큰이 유효하지 않습니다
AUTH_002	401	토큰이 만료되었습니다
AUTH_003	403	권한이 없습니다
AUTH_004	409	이미 존재하는 이메일입니다
AUTH_005	409	이미 존재하는 전화번호입니다
AUTH_006	409	이미 존재하는 닉네임입니다

9.2 사용자 관련 에러

코드	HTTP 상태	설명
USER_001	404	사용자를 찾을 수 없습니다
USER_002	400	잘못된 비밀번호입니다
USER_003	400	입력 데이터가 유효하지 않습니다

9.3 시설 관련 에러

코드	HTTP 상태	설명
FACILITY_001	404	시설을 찾을 수 없습니다
FACILITY_002	403	시설 등록 권한이 없습니다
FACILITY_003	400	잘못된 시설 정보입니다

10. API 사용 예시

10.1 회원가입 → 로그인 → 시설 검색 플로우

```
// 1. 회원가입
const signupResponse = await fetch('/api/v1/auth/register', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    email: 'user@example.com',
    phone: '010-1234-5678',
    password: 'password123!',
    name: '홍길동',
    nickname: '길동이',
    birthDate: '1990-01-01',
    gender: 'M',
    userType: 'USER',
    termsAgreed: {
      service: true,
      privacy: true,
      marketing: false
    }
  })
});

// 2. 로그인
const loginResponse = await fetch('/api/v1/auth/login', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    loginId: 'user@example.com',
    password: 'password123!'
  })
});

const loginData = await loginResponse.json();
const accessToken = loginData.data.accessToken;

// 3. 시설 검색
const facilitiesResponse = await fetch('/api/v1/search/facilities?keyword=헬스
&latitude=37.5665&longitude=126.9780', {
  headers: {
    'Authorization': `Bearer ${accessToken}`
  }
});
```

작성일: 2025년 10월

버전: 2.0

작성자: 김진석

기반 문서: DB 설계서

개발 범위: 1차 개발(MVP) - 공공데이터 기반 읽기 전용, 예약/리뷰/찜/문의/알림/커뮤니티/편집 기능 제외